

Anmelderin:
softel GmbH
Anrather Str. 21
47877 Willich

5

Verfahren zur Bearbeitung von CDR-Informationen

10 Beschreibung

Die Erfindung betrifft ein Verfahren zur Analyse und Bearbeitung, insbesondere Korrektur einer Menge von Call 15 Detail Records (CDRs).

Gebiet der Erfindung:

Verschiedenste Netzelemente von Mobilfunkbetreibern oder anderen Dienstleistern (Finanzinstitute, Energieunternehmen, Internetserviceprovider) erzeugen servicerelevante Daten, so genannte Call Detail Records (CDRs), die zum Großteil in die Rechnungserstellung einfließen. Die meisten CDRs liegen in proprietären binären Formaten vor und werden von so genannten Mediation Devices u. a. eingesammelt, aggregiert und auf interne oder standardisierte Datenformate zur Weiterverarbeitung innerhalb einer Abrechnungskette abgebildet.

Die Abrechnungskette, die die Verarbeitung dieser servicerelevanten Abrechnungsdaten vornimmt, ist ein hoch komplexes, mehrstufiges Softwaresystem, das meist von verschiedenen Hard- und Softwarelieferanten realisiert wurde. Durch die Komplexität und die ständig anfallenden Änderungswünsche kommt es permanent vor, dass ein gewisser

Teil der Abrechnungsdaten fehlerbehaftet ist und dadurch nicht abgerechnet werden kann. Unabhängige Branchenkennzahlen sprechen von ca. 2-5% Verlusten.

Aufgabe der Erfindung:

- 5 Aufgabe der Erfindung ist es, ein Analyse- und/oder Korrektursystem bereitzustellen, mit dem es möglich ist, abrechnungsrelevante Daten allgemeingültig zu analysieren, Fehler zu finden, diese zu reparieren und die reparierten Daten der Abrechnungskette wieder zuzuführen.
- 10 Diese Aufgabe wird durch die Erfindungen mit den Merkmalen der unabhängigen Ansprüche gelöst. Vorteilhafte Weiterbildungen der Erfindungen sind in den Unteransprüchen gekennzeichnet.

Dies wird im Wesentlichen dadurch erreicht, dass in einer Konfigurationsdatei Strukturinformationen hinterlegt sind, die die Grammatik der CDRs wiedergeben. Diese Dateien werden eingelesen und es wird eine relationale Datenbank auf der Grundlage der Grammatik erzeugt. Hierbei ist insbesondere zu beachten, dass hierarchische Grammatikkonstrukte, die Rekursionen erlauben, durch eine 1:N Verknüpfung dargestellt werden. Nach dem Erstellen der Datenbank werden die CDRs in die Datenbank eingelesen, um dann mit frei definierbaren oder vorgegebenen SQL-Queries nach Fehlern zu suchen, die ggf. auch automatisch korrigiert werden können. Nach der Korrektur werden die CDRs aus der Datenbank auf der Basis der Strukturinformationen ausgelesen und im ursprünglichen Format abgespeichert.

Das System stellt hierdurch Funktionen zur Analyse und Behebung von Fehlern in vom Mediation Device gesammelten oder erzeugten CDRs mit abrechnungsrelevanten Daten aus beliebigen Datenquellen zur Verfügung. Dazu ist es nötig, CDRs-diversen Formate für schnelle Suche zwischenspeichern, fehlerhafte zu erkennen und zu analysieren. Das erfindungsgemäße

Korrekturverfahren ist zur Unterstützung von Massenupdates ausgelegt.

Fehler in den CDRs haben die Eigenschaft, mehrfach und über längere Zeit aufzutreten; deshalb ist eine Wiederholbarkeit von Analysen und Korrekturmaßnahmen verfügbar. Das erfindungsgemäße System ermöglicht ebenfalls, eine lückenlose Rückverfolgung und Wiederverwendung aller gemachten Korrekturen, indem protokolliert wird, wer zu welchem Zeitpunkt welche Änderung vorgenommen hat. Dies wird durch das

10 Anlegen von Kopien erreicht, auf denen alle Änderungsoperationen durchgeführt werden. Des Weiteren bietet das System Revisionssicherheit, d. h. dass zu jedem späteren Zeitpunkt mit den gleichen Eingangsdaten die Korrekturen erneut durchgeführt werden können und zu den gleichen
15 Endergebnissen führen.

Eine weitere Systemeigenschaft ist die Unterstützung der Benutzer mit Anzeigen und Sortiermöglichkeiten zur optimalen Darstellung von Suchergebnissen und von vorgenommenen Korrekturen.

20 So stehen in der bevorzugten Ausführungsform folgende Systemfunktionen zur Verfügung: Eingabefunktionen und Zwischenspeicherung von Modifikationen, dynamische Bestimmung der Strukturdefinition der Eingabe-CDRs, Abbildung der Strukturdefinition auf die zwischenspeichernde Datenbank,
25 leistungsfähige Eingabeschnittstellen in die Datenbank, Laden in eine relationale Datenbankinstanz, Import von durch Wildcards identifizierten Dateien, Importformatierung mittels der Strukturdefinitionen, Wiederaufsetzen bei defekten Bereichen, Header- und Trailertest, insbesondere für das TAP3-
30 Format, Behandlung von defekten Bereichen in CDR-Dateien nach definierten Mustern, Suchfunktionen, Definition von Ergebnismengen basierend auf SQL- oder PLSQL-Abfragesprache, weitere Verfeinerung von Ergebnismengen mit PL/SQL, Sortierung

und Gruppierung von Ergebnismengen mit Standard-SQL, Export- und Dumpmöglichkeit der Oracle-Datenbank, so dass die Daten für die Weiterverarbeitung in das Billing-System übergeben werden können. Die folgenden Bearbeitungsfunktionen stehen

- 5 dabei zur Verfügung: Bearbeitung der Ergebnismenge mit Standard SQL/PLSQL-Update-Anweisungen, Bearbeitung der Ergebnismenge mit PL/SQLDatenbank-Prozeduren, Zuordnung einer Korrekturnummer zu einer Ergebnismenge (Revision Control Mode), Bearbeitung der Ergebnismenge mit PL/SQL-Datenbank-10 Prozeduren innerhalb der Benutzeroberfläche, Freigabe oder Rücknahme einer Korrektur, Änderungsprotokollierung, Protokollierung von WER macht WANN, WAS und WIE und der Ausgaben mit Zuordnung zu den Korrekturnummern, Ausgabefunktionen, Zuordnung von Dateinamen zu den Ausgaben, 15 Vermerk des korrekten Endes einer Ausgabe, Ausgabemöglichkeit der korrigierten CDRs und optional der ursprünglichen CDRs oder kompletter Eingabefiles, Erzeugung des zum EingabefORMAT identischen Ausgabeformats mit den vorhandenen Strukturdefinitionen, Auflistung der ins System geladenen CDR-20 Dateien. Zur vereinfachten Bearbeitung steht eine Benutzeroberfläche zur Verfügung, die intuitiv zu bedienen ist. Es wird darauf hingewiesen, dass das System auch aus Unterkombinationen der o. g. Funktionen bestehen kann.

Im Folgenden wird die Erfindung anhand von

- 25 Ausführungsbeispielen näher erläutert, die in den Figuren schematisch dargestellt sind. Gleiche Bezugsziffern in den einzelnen Figuren bezeichnen dabei gleiche Elemente. Im Einzelnen zeigt:

- 30 Fig. 1 den strukturierten Aufbau der vorliegenden Erfindung.

Die zu analysierenden CDR-Tagesproduktionen-Daten liegen in einem Arbeitsverzeichnis in Dateiform vor. Anhand des Dateinamens oder durch Benutzerangabe werden das Recordformat und die Formatversion der CDRs bestimmt. Innerhalb einer Datei dürfen nur CDRs eines Recordformates in einer Version enthalten sein. Welche CDR-Dateien zu laden sind, wird vom Benutzer festgelegt.

Anhand des erkannten Recordformates werden mit Hilfe der Strukturdateien Datenbanktabellen angelegt. Die Daten können dann mit dem formatabhängigem Decoder (GSM, TAP3, Mediation, u. a.) decodiert und in die Datenbank geladen werden.

Sowohl die Funktionsweise der Decoder, Encoder als auch die automatisierte Abbildung der Datenbeschreibungssprache auf eine relationale Datenbank wird in der bevorzugten Ausführungsform über die Strukturdatei bestimmt.

Die Decoder haben in der bevorzugten Ausführungsform Methoden, um defekte Bereiche innerhalb von CDR-Dateien zu erkennen, wieder aufzusetzen, zu überspringen und Korrekturen vorzubereiten. Zusätzlich sind die Decoder mit Testfunktionen und Dumpmöglichkeiten ausgestattet.

Durch die Verwendung einer relationalen Datenbank stehen alle Möglichkeiten der SQL-Sprache, der PL/SQL-prozeduralen Datenbanksprache und Programme mit Dialog im Zeilenmodus zur Verfügung. So können weiterhin 3rd-Party-Tools benutzt werden, um auf die Tabellen zuzugreifen, zu suchen und Veränderungen vorzunehmen.

Die in der bevorzugten Ausführungsform vorliegende Benutzeroberfläche beinhaltet in ihrem Bearbeitungsteil Benutzerdialoge zum Suchen, Darstellen, Korrigieren und Protokollieren von Ergebnissen. Für größtmögliche Flexibilität können in der Oberfläche SQL-Anweisungen und PL/SQL-Datenbankprogramme definiert und für spätere Wiederverwendung

abgespeichert werden. Zur Darstellung und Manipulation ist ein CDR-Editor integriert. Ist ein Fehler gefunden und die betroffenen Daten selektiert worden, so können diese innerhalb der Oberfläche mit Massenupdatefunktionen korrigiert werden.

- 5 Alle Änderungen werden persistent und tagesgenau protokolliert und stehen für Revisionszwecke zur Verfügung, indem auf Kopien gearbeitet wird. Auch die Oberfläche kann ihr Layout, wie z. B. die Anordnung der Fenster, speichern und wieder verwenden.
- 10 Die CDR-Strukturdateien sind eine Abbildung der CDR-Spezifikationen bzw. Grammatik. Alle Feldbeschreibungen und die zugehörigen Wertebereiche für verschiedene Netzelemente sind in ihnen abgelegt. Sie werden direkt aus den Spezifikationen der Netzwerkausrüster erstellt. Zu einer
- 15 Version eines Formats existiert immer nur eine Strukturdatei. Alle Module greifen auf diese eine Strukturdatei zur Formatbeschreibung zu. Dies hat den Vorteil, dass man bei zu erwartenden Änderungen an CDR-Formaten von Netzwerkelementen diese nur einmalig an zentraler Stelle in die Strukturdateien
- 20 einarbeiten braucht. Diese Änderungen stehen dann sofort allen Modulen zur Verfügung.

Die CDR-Strukturen werden in der Datenbank auf Tabellen abgebildet.

- 25 Es werden in der bevorzugten Ausführungsform folgende Arten von Strukturdateien unterschieden:

1. ASN.1-Strukturdatei, die ASN.1 codierte Datenformate beschreibt.
2. Nicht-ASN.1-Strukturdatei, die in einer erfindungsgemäßen Datenbeschreibungssprache vorliegt und alle nicht ASN.1-
- 30 Datenformate beschreibt.

Zum Verarbeiten dieser Formate sind zusätzlich die formatspezifischen CDR-Decoder sowie CDR-Encoder notwendig. Die Strukturdateien können somit einer fort dauernden Pflege und Erweiterung für neue Netzwerkelemente und neue

5 Softwareversionen unterliegen.

Die CDR-Decoder konvertieren alle oftmals im binären Format vorliegenden CDRs in ein Datenbank lesbares Format, vorzugsweise in das ASCII-Format, das so genannte FAF-Format.

In diesem Format werden die Feldnamen, Datenstrukturen und

10 Datentypen dargestellt. Die Decoder sind vorzugsweise

Batchprogramme. Die zu dekodierenden Formate werden

vorzugsweise automatisch über Namenskonventionen für

Eingabedateien bestimmt. Andere Analysen zur Bestimmung des

Formates sind jedoch denkbar. Aus dem Dateinamen sollten der

15 notwendige Netzelementtyp sowie die zugehörige Versionsnummer

des Formates hervorgehen. Format und Version können jedoch

auch abweichend von der Namenskonvention vorgegeben werden.

Der CDR-Decoder-Output (FAF-Format) wird, vorzugsweise ohne Zwischenspeicherung, als direkter Input für das

20 Datenbankinterface verwendet. Dadurch wird das File-system der

Rechner nur zum Lesen der Eingabedaten benötigt. Das

aufwendige Speichern von Zwischenergebnissen entfällt.

Um eine möglichst große Anzahl von CDRs aus Dateien mit

defekten Bereichen zu extrahieren, beinhalten die Decoder

25 Funktionen zum Wiederaufsetzen während des Lesens einer

Eingabedatei. Es werden Pattern (Suchmuster) festgelegt, die

den Anfang von neuen Records beschreiben. Mit diesen

Informationen versuchen die Decoder defekte Bereiche mit

minimalem Verlust zu überspringen. Die Anzahl der

30 Aufsetzversuche wird vorzugsweise in der

Verwaltungsinformation zur CDR-Datei vermerkt. Je Format und

Version ist ein Maximum von Aufsetzversuchen definiert. Nach

Erreichen dieses Maximums wird das Decodieren der Eingabedatei abgebrochen.

Zur manuellen Behandlung von fehlerhaften Bereichen in CDR-Dateien sind Methoden in den Decodern enthalten, die

5 fehlerhafte Bereiche von CDR-Dateien in unstrukturierter Form in die Datenbank schreiben. Durch Schneiden, d. h. durch Setzen von Anfangs- und Endmarken, werden korrupte Bereiche in den Eingabedaten gekennzeichnet. Die Benutzeroberfläche erlaubt, in einer bevorzugten Ausführungsform einzelne Bytes
10 in unstrukturierten Bereichen in hexadezimaler Notation zu verändern.

CDRs, die geladen wurden, sind nur lesbar, um unabsichtliche Veränderungen ohne begonnene Korrektur zu vermeiden.

Testfunktionen zur Prüfung der inhaltlichen Integrität einer
15 CDR-Datei können beim Import pro Format automatisch oder manuell ausgeführt werden.

SQL-Abfragen beinhalten auch die Prüfung auf den NULL-Wert oder den 'Leer-Wert', um festzustellen, ob eine Spalte gefüllt ist. Um die Abfragen noch weiter verfeinern zu können, sind
20 SQL-, PL/SQL-Funktionen verfügbar, die in der WHERE-Klausel verwendet werden können.

Suchabfragen werden über das CDR-Select-Kommando definiert und bauen hierarchisch aufeinander auf. Das bedeutet, dass jedes CDR-Select-Kommando auf einem anderen CDR-Select basiert und
25 die Suchmenge über eine UND-Verknüpfung immer weiter eingeschränkt wird. Jede so definierte Suchmenge erhält vorzugsweise einen eigenen Namen, welcher als Referenz für weitere CDR-Selects benutzt werden kann. Die vollständigen SQL-Queries werden vom System automatisch erzeugt. Die
30 ursprünglich geladene Gesamtmenge erhält den Namen Initial.

So ist z.B. eine automatische Query ein Header Trailer Check und Update am Beispiel vom TAP3-Standard.

Im Header Check wird beispielsweise für folgende Referenzdaten die korrekte Verwendung innerhalb der Call Detail Records überprüft.

- 5 1. taxes
- 2. discounts
- 3. exchange rates
- 4. utc time offsets
- 5. rec entities
- 10 6. vas
- 7. message descriptions
- 8. called number analysis

Für diese Daten sind innerhalb des TAP3 Formates

- 15 Unterstrukturen definiert. D. h. im File sollen z. B. alle vorkommenden Taxes (Taxcodes, Taxrate, ...) definiert sein. Innerhalb der eigentlichen CDRs sollen nur diese Taxes referenziert werden.
- 20 Werden Taxes referenziert, die nicht definiert sind, wird ein Warning ausgegeben. Eine automatische Korrektur kann hier nicht stattfinden, da nicht entschieden werden kann, ob die Definition des Taxes fehlt, oder ob die Referenz im CDR fehlerhaft ist. Der PAT-Benutzer muss entweder die Tax in die Referenzstruktur hinzufügen (CDR INSERT) oder den Taxcode des
- 25 CDR ändern (CDR UPDATE).

Der Trailer-Check überprüft, ob die im Trailer hinterlegten Summen mit den tatsächlich summierten Werten der CDRs

übereinstimmen. Ausnahme ist der früheste und späteste „call timestamp“.

Folgende Werte werden kontrolliert.

- 5 1. earliest and latest call timestamp
2. total charges
3. total taxes
4. total discounts
5. call event detail counts

10 Der Trailer wird nach jeder Änderung der CDRs (CDR INSERT, CDR UPDATE, CDR DELETE) und vor jedem export neu berechnet

Weitere automatische Queries sind denkbar und individuell auf dem SQL-Standard zu definieren.

Zur Sicherstellung der Konsistenz können Korrekturen

15 vorzugsweise nur auf Ergebnismengen der Suche angewendet werden. Dazu ist es notwendig, eine Revision, d. h. eine Kopie, zu eröffnen. Das Neuerstellen einer Revision bewirkt, dass die Ergebnismenge in einen temporären, schreibbaren Datenbankbereich kopiert wird, in dem Änderungen mittels des

20 CDR-Update-Kommandos durchgeführt werden können. Durch diese Vorbereitung wird jeder veränderte CDR mit seiner CDR-ID innerhalb der Datenbank vermerkt.

Da über die aus den Strukturdateien abgeleiteten Tabellendefinitionen alle Spalten optional sind, kann jede

25 Spalte auf NULL gesetzt werden. Zusätzlich besteht die Möglichkeit, Felder abhängig von deren Datentyp auf einen 'Leer-Wert' zu setzen.

In der bevorzugten Ausführungsform wird das Löschen von CDRs indirekt bereitgestellt. Jede Tabellendefinition wird um ein Löschkennzeichen erweitert. Eine UPDATE-Anweisung setzt dieses Kennzeichen für zu löschen CDRs.

5 CDR-Dateien, in denen CDRs geändert wurden, werden als geändert markiert. In Abhängigkeit der Form des Exports können z. B. nur geänderte CDR-Dateien exportiert werden.

Korrekturen finden vorzugsweise nur auf Ergebnismengen (Revisionen) statt, da sonst eine Wiederholbarkeit nicht mehr

10 gegeben ist. Um Revisionssicherheit zu erlangen, werden deshalb Veränderungen, vorzugsweise über SQL-UPDATE-Anweisungen oder über PL/SQL-Programme manuell oder automatisch vorgenommen. PL/SQL-Programme ermöglichen, Werte zu speichern und Platzhalter in UPDATE-Anweisungen mit Werten 15 zu besetzen, bevor die Anweisung ausgeführt wird.

Korrekturen werden vorzugsweise nur in Verbindung mit einer Korrekturnummer vorgenommen, die einer Ergebnismenge zugeordnet wird. Zusammen mit dieser Nummer werden die Korrekturanweisungen und alle weiteren Aktionen abgespeichert.

20 Protokolliert wird 'wer', 'was', 'wann', 'warum' und eine Summenstatistik. Jeder veränderte CDR wird mit Korrekturnummer und CDR-ID innerhalb der Datenbank protokolliert. Dieses ist vorteilhaft, um mehrfache Korrekturen des gleichen CDRs feststellen zu können. Der Export von Korrekturen wird 25 vorzugsweise von dieser Protokolltabelle beeinflusst. Eine Lösung zum Duplikatsproblem ist in der Ausgabefunktion beschrieben.

Korrekturen werden vorzugsweise mit einem Namen und einer Beschreibung versehen und können mit einer jeweils neuen

30 Korrekturnummer aufgerufen und wiederholt werden. Zur Beschleunigung kann verringertes Logging gesetzt werden. Bei Wiederverwendung einer Korrektur werden die 'was und warum'-Felder der Protokollierung mit den letzten Werten vorbesetzt.

Die geänderten CDRs werden aus der Datenbank exportiert und mit den formatspezifischen Encodern unter Verwendung der Strukturdefinition in das binäre Zielformat konvertiert. Die so geänderten Daten sind für weitere Verarbeitungsschritte in 5 vorgegebenen Zielverzeichnissen zusammen mit der Protokollierung abgelegt. Die Ausgabe selbst bietet einige Optionen, wie z. B. die Ausgabe der ursprünglichen CDRs.

Logisch sind folgende Arten von Ausgaben zu unterscheiden.

1. Die Ausgabe beschränkt auf die korrigierten CDRs in eine 10 neu erstellte Datei.
2. Die Ausgabe beschränkt auf die korrigierten CDRs unter Beibehaltung der ursprünglichen Dateinamen und
3. Die Ausgabe von kompletten Daten mit dem ursprünglichen 15 Dateinamen, in die die geänderten und die nicht von den Änderungen betroffenen Daten in ihrer ursprünglichen Reihenfolge ausgegeben werden.

Die erste und zweite Art wird verwendet, wenn nur Teile einer Datei mit Fehlern behaftet waren, die Datei an sich aber im Abrechnungssystem verarbeitet wurde. Die dritte Art kann 20 verwendet werden, um vom Abrechnungssystem als ganzes abgewiesene CDR-Dateien neu zu verarbeiten.

Patentansprüche

1. Verfahren zur Analyse und/oder Bearbeitung von einer
5 Menge von Call Detail Records (CDRs),
 - mit einem Initialisierungsschritt, bei dem aus einem ersten Speicherbereich Strukturinformationen der Call Detail Records geladen werden und anhand dieser eine relationale Datenbank aufgebaut wird, die bei hierarchischen Strukturinformationen mindestens aus zwei miteinander rekursiv in Relation stehenden Tabellen besteht, die die Hierarchie abbilden,
 - mit einem weiteren Schritt, bei dem aus einem zweiten Speicherbereich die vorzugsweise binär vorliegende Menge von Call Detail Records in ein lesbares Format der Datenbank umgewandelt wird, insbesondere in das ASCII-Format,
 - mit einem weiteren Schritt, bei dem die lesbaren Call Detail Records entsprechend der Strukturinformationen in die Datenbank geladen werden, die im Initialisierungsschritt erzeugt wurde,
 - mit einem weiteren Schritt, bei dem SQL-Queries und/oder Datenbankprozeduren verwendet werden, um fehlerhafte Call Detail Records zu bestimmen, die geändert werden können,
 - mit einem weiteren Schritt, bei dem mit Hilfe der Strukturinformationen und dem Aufbau der Datenbank die Call Detail Records in einen dritten Speicherbereich zurückgespeichert werden, der im Aufbau der ursprünglichen Menge an Call Detail Records entspricht.

2. Verfahren nach dem vorhergehenden Anspruch, **dadurch gekennzeichnet**, dass für jeden Typ in der Strukturdefinition eine Tabelle angelegt wird.

5 3. Verfahren nach einem oder mehreren der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, dass bei der Auflösung von hierarchischen Datenstrukturen solange normalisiert wird, bis keine 10 nicht beschränkte Wiederholungsgruppe vorliegt, wobei im Falle von nicht beschränkten Wiederholungsgruppen rekursiv eine 1:N Relation zwischen einer neuen Tabelle aufgebaut wird, die wiederum normalisiert die Wiederholungsgruppe aufnimmt.

15 4. Verfahren nach einem oder mehreren der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, dass der Decoder zum Einlesen der Menge von Call Detail Records Binärdaten nach Fehlern, insbesondere nach Protokollfehlern, durchsucht, um diese zu 20 markieren, und um ggf. den Einleseschritt neu anzusetzen.

5. Verfahren nach dem vorhergehenden Anspruch, **dadurch gekennzeichnet**, dass die fehlerhaften Bereiche in 25 eine ausgewiesene Datenstruktur geschrieben werden.

6. Verfahren nach einem oder mehreren der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, dass benutzerspezifische Erweiterungen, insbesondere 30 SQL-Select-Statements, für die Suche von defekten Call Detail Records individuell abspeicherbar sind.

7. Verfahren nach einem oder mehreren der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, dass bei einer Änderung ein Duplikat des Call Detail 35

Records angelegt wird
und/oder
eine Protokollierung stattfindet, indem vorzugsweise
der Bearbeiter und/oder der Zeitraum und/oder eine
5 Protokollnummer vergeben wird, und/oder
bei einer Löschung ein Löschkennzeichen vergeben
wird.

8. Verfahren nach einem oder mehreren der
10 vorhergehenden Ansprüche, **dadurch gekennzeichnet**,
dass die Ausgabe durch einen Encoder erfolgt, der
auf der Basis der Strukturinformationen und der
Datenbank die Call Details Records im ursprünglichen
Einleseformat ausgibt, das vorzugsweise binär ist.

15 9. Verfahren nach einem oder mehreren der
vorhergehenden Ansprüche, **dadurch gekennzeichnet**,
dass die Eingabe und/oder Ausgabe in eine Datei
erfolgt, die vorzugsweise im Binärformat die Menge
20 von Call Details Records speichert.

10. Verfahren nach einem oder mehreren der
vorhergehenden Ansprüche, **dadurch gekennzeichnet**,
dass automatische Korrekturen bei dem Vorliegen
25 bestimmbarer Mustern durchgeführt werden.

30 11. Verfahren nach einem oder mehreren der
vorhergehenden Ansprüche, **dadurch gekennzeichnet**,
dass eine Änderung der Strukturdatei sich auf den
Aufbau der Datenbanktabellen und/oder auf die
Funktion des Encoders und/oder auf die Funktion des
Decoders auswirken.

35 12. Verfahren nach einem oder mehreren der
vorhergehenden Ansprüche, **dadurch gekennzeichnet**,

dass die Datenbank dynamisch bei jedem erneuten Einlesen einer Datei erzeugt wird oder einmalig für jeden Strukturtyp, der durch eine Strukturinformation bestimmt ist.

5

13. Verfahren nach einem oder mehreren der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, dass Views in Abhängigkeit der Strukturinformationen für eine Datenbank generiert werden.

10

14. Software für einen Computer, **dadurch gekennzeichnet**, dass ein Verfahren nach einem oder mehreren der vorhergehenden Ansprüche für einen Computer implementiert ist.

15

15. Datenträger für einen Computer, **gekennzeichnet durch** die Speicherung einer Software nach dem vorhergehenden Softwareanspruch.

20

16. Computersystem, **gekennzeichnet durch** eine Einrichtung, die den Ablauf eines Verfahrens nach einem oder mehreren der vorhergehenden Verfahrensansprüche erlaubt.

Anmelderin:
softel GmbH
Anrather Str. 21
47877 Willich

Verfahren zur Bearbeitung von CDR-Informationen

Zusammenfassung

Die Erfindung betrifft ein Verfahren zur Analyse und Bearbeitung, insbesondere Korrektur einer Menge von Call Detail Records (CDRs). Dies wird im Wesentlichen dadurch erreicht, dass in einer Konfigurationsdatei Strukturinformationen hinterlegt sind, die Grammatik der CDRs wiedergibt. Diese Dateien werden eingelesen und es wird eine relationale Datenbank auf der Grundlage der Grammatik erzeugt. Hierbei ist insbesondere zu beachten, dass hierarchische Grammatikkonstrukte, die Rekursionen erlauben, durch eine 1:N Verknüpfung dargestellt werden. Nach dem Erstellen der Datenbank werden die CDRs in die Datenbank eingelesen, um dann mit vorgegebenen SQL-Queries nach Fehlern zu suchen, die ggf. auch automatisch korrigiert werden können. Nach der Korrektur werden die CDRs aus der Datenbank auf der Basis der Strukturinformationen ausgelesen und im ursprünglichen Format abgespeichert.

(Fig. 1)

Architektur der Professionalversion

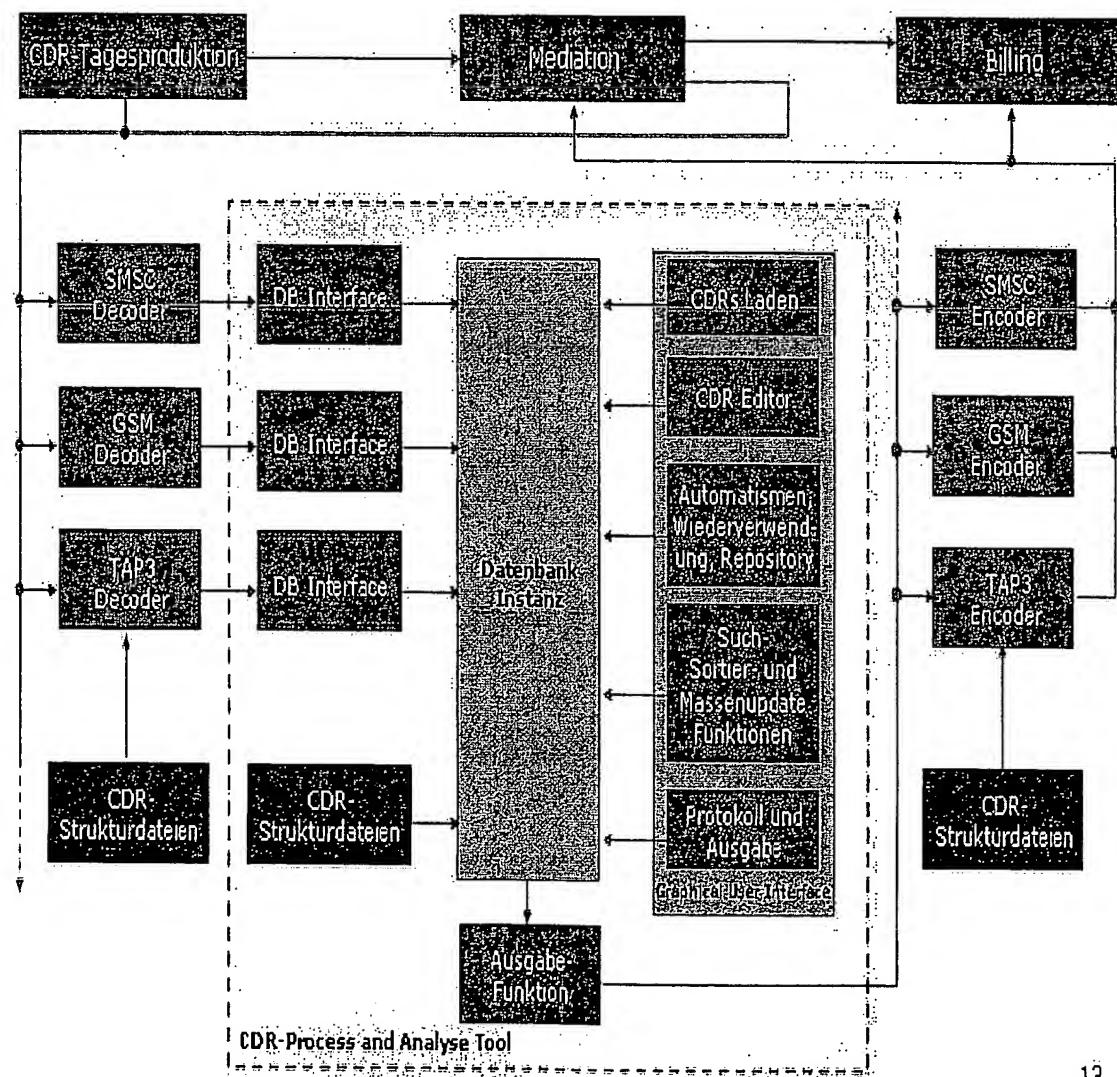


Fig. 1

BEST AVAILABLE COPY